MEGALODON: Efficient LLM Pretraining and Inference with Unlimited Context Length

Xuezhe Ma[π*] **Xiaomeng Yang**[μ*] **Wenhan Xiong**[μ] **Beidi Chen**[κ] **Lili Yu**[μ]

Hao Zhang[δ] **Jonathan May**[π] **Luke Zettlemoyer**[μ] **Omer Levy**[μ] **Chunting Zhou**[μ*]

[μ]AI at Meta
[π]University of Southern California
[κ]Carnegie Mellon University
[δ]University of California San Diego

Abstract

The quadratic complexity and weak length extrapolation of Transformers limits their ability to scale to long sequences, and while sub-quadratic solutions like linear attention and state space models exist, they empirically underperform Transformers in pretraining efficiency and downstream task accuracy. We introduce MEGALODON, an neural architecture for efficient sequence modeling with unlimited context length. MEGALODON inherits the architecture of MEGA (exponential moving average with gated attention), and further introduces multiple technical components to improve its capability and stability, including *complex exponential moving average (CEMA)*, *timestep normalization* layer, *normalized attention* mechanism and *pre-norm with two-hop residual* configuration. In a controlled head-to-head comparison with LLAMA2, MEGALODON achieves better efficiency than Transformer in the scale of 7 billion parameters and 2 trillion training tokens. MEGALODON reaches a training loss of 1.70, landing mid-way between LLAMA2-7B (1.75) and 13B (1.67). The improvements of MEGALODON over Transformers are robust throughout a range of benchmarks across different tasks and modalities. **Code**: https://github.com/XuezheMax/megalodon

1 Introduction

In many real-world applications, such as multi-turn conversation, long-document comprehension, and video generation, large language models (LLMs) must efficiently process long sequential data, understand internal long-range dynamics, and generate coherent output. The Transformer architecture (Vaswani et al., 2017), despite its remarkable capabilities, faces challenges with quadratic computational complexity and limited inductive bias for length generalization, making it inefficient for long sequence modeling (Wang et al., 2024; Zhou et al., 2024). Even with recently proposed distributed attention solutions (Li et al., 2023b; Liu et al., 2024), computing a single training step of a 7B parameter model over a 1M-token sequence is more than 100 times slower than performing the equivalent computation using 256 separate sequences of 4K tokens each.

Techniques like efficient attention mechanisms (Tay et al., 2020; Ma et al., 2021) and structured state space models (Gu et al., 2022a; Poli et al., 2023; Gu and Dao, 2023) have been introduced to overcome these limitations, aiming to enhance scalability and performance. However, the practical application of these methods still falls short of Transformers (Tay et al., 2022; Gu and Dao, 2023). This work introduces an unlimited context model that outperforms the canonical Transformer architecture on real-world language modeling.

*Equal Contribution. Correspondence to chuntinz@meta.com

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2294

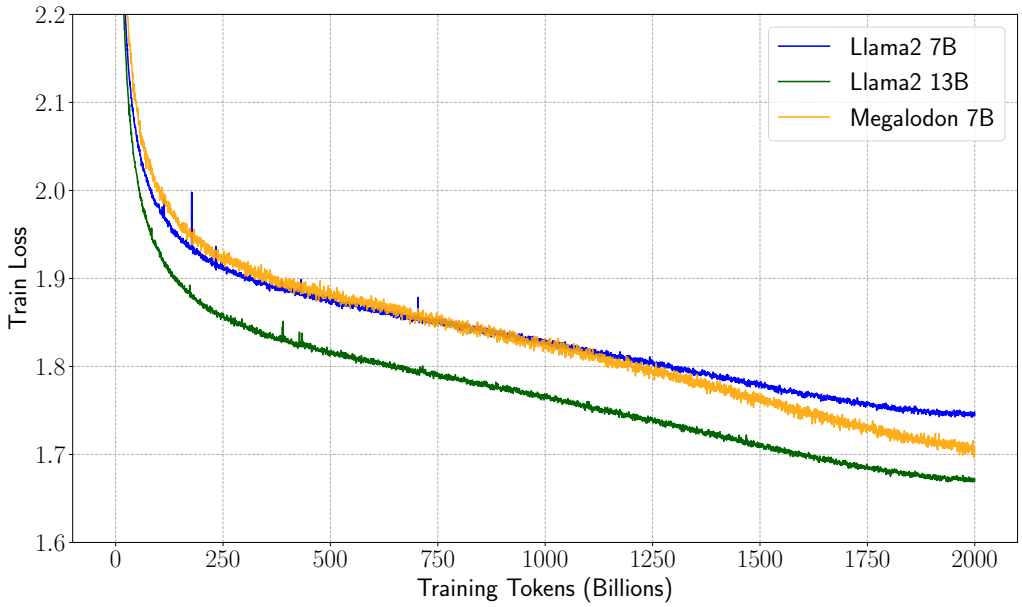


Figure 1: **Negative log-likelihood** for MEGALODON-7B, LLAMA2-7B and LLAMA2-13B.

Table 1: **Performance on standard academic benchmarks**, compared to open-source base models. We reported model size, context length and total data tokens during model pretraining. – indicates that the number was not reported in the original paper.

Model	Size	Tokens	Context	MMLU	BoolQ	HellaSw	PIQA	SIQA	WinoG	Arc-e	Arc-c	NQ	TQA
Mamba	3B	0.6T	2K	26.2	71.0	71.0	78.1	–	65.9	68.2	41.7	–	–
RWKV	7B	1.1T	4K	–	–	70.8	77.3	–	68.4	74.9	46.1	–	–
MPT	7B	1T	4K	26.8	75.0	76.4	80.6	48.5	68.3	70.2	42.6	20.8	50.4
Mistral	7B	–	16K	60.1	**83.2**	**81.3**	**82.2**	47.0	**74.2**	80.0	**54.9**	23.2	62.5
Gemma	8B	6T	8K	**64.3**	**83.2**	81.2	81.2	**51.8**	72.3	**81.5**	53.2	23.0	63.4
LLAMA2	13B	2T	4K	54.8	81.7	80.7	80.5	50.3	72.8	77.3	49.4	**31.2**	**65.1**
LLAMA2	7B	2T	4K	45.3	77.4	77.2	78.8	48.3	69.2	75.2	45.9	25.7	58.5
MEGALODON	7B	2T	32K	49.8	80.5	77.5	80.1	49.6	71.4	79.8	53.1	25.7	60.5

We introduce MEGALODON, an improved MEGA architecture (Ma et al., 2023), which harnesses the gated attention mechanism with the classical exponential moving average (EMA) (Hunter, 1986) approach (§2). To further improve the capability and efficiency of MEGALODON on large-scale long-context pretraining, we propose multiple novel technical components. First, MEGALODON introduces the *complex exponential moving average (CEMA)* component, which extends the multi-dimensional damped EMA in MEGA to the complex domain (§3.1). Then, MEGALODON proposes the *timestep normalization* layer, which generalizes the group normalization layer (Wu and He, 2018) to auto-regressive sequence modeling tasks to allow normalization along the sequential dimension (§3.2). To improve large-scale pretraining stability, MEGALODON further proposes *normalized attention* (§3.3), together with *pre-norm with two-hop residual* configuration by modifying the widely-adopted pre- and post-normalization methods (§3.4). By simply chunking input sequences into fixed blocks, as is done in MEGA-chunk (Ma et al., 2023), MEGALODON achieves linear computational and memory complexity in both model training and inference.

Empirically, we demonstrate the potential of MEGALODON as a general architecture for modeling long sequences, by evaluating its performance across multiple scales of language modeling, as well as downstream domain-specific tasks. Through a direct comparison with LLAMA2, while controlling for data and compute, MEGALODON-7B significantly outperforms the state-of-the-art variant of Transformer used to train LLAMA2-7B (Touvron et al., 2023) on both training perplexity (Figure 1) and across downstream benchmarks (Table 1). Evaluation on long-context modeling, including perplexity in various context lengths up to 2M and long-context QA tasks in Scrolls (Parisotto et al., 2020) prove MEGALODON's ability to model sequences of unlimited length. Additional experimental results on small/medium-scale benchmarks, including LRA (Tay et al., 2021), ImageNet (Deng et al., 2009), Speech Commands (Warden, 2018), WikiText-103 (Merity et al., 2017) and PG19 (Rae et al., 2019), demonstrate the robust improvements of MEGALODON across scales and modalities.

https://doi.org/10.52202/079017-2294

2 Background: Moving Average Equipped Gated Attention (MEGA)

In this section, we setup notations, briefly review the key components in the MEGA architecture (Ma et al., 2023), and discuss the existing problems in MEGA.

Following the notations in MEGA, we use $\boldsymbol{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\} \in \mathbb{R}^{n \times d}$ to denote the input and output sequences with length n, and assume the representations of the input and output sequences have the same dimension d.

2.1 Multi-dimensional Damped EMA

MEGA embeds an EMA component into the calculation of the attention matrix to incorporate inductive biases across the timestep dimension. Concretely, the multi-dimensional damped EMA first expands each dimension of the input sequence $\boldsymbol{X}$ individually into h dimensions via an expansion matrix $\boldsymbol{\beta} \in \mathbb{R}^{d \times h}$, then applies damped EMA to the h-dimensional hidden space. Formally, for each dimension $j \in \{1, 2, \ldots, d\}$:

$$\begin{aligned} \mathbf{u}_t^{(j)} &= \boldsymbol{\beta}_j \mathbf{x}_{t,j} \\ \mathbf{h}_t^{(j)} &= \boldsymbol{\alpha}_j \odot \mathbf{u}_t^{(j)} + (1 - \boldsymbol{\alpha}_j \odot \boldsymbol{\delta}_j) \odot \mathbf{h}_{t-1}^{(j)} \\ \mathbf{y}_{t,j} &= \boldsymbol{\eta}_j^T \mathbf{h}_t^{(j)} \end{aligned} \tag{1}$$

where $\mathbf{u}_t^{(j)} \in \mathbb{R}^h$ is the expanded h-dimensional vector for the j-th dimension at timestep t. $\boldsymbol{\alpha} \in (0,1)^{d \times h}$, $\boldsymbol{\delta} \in (0,1)^{d \times h}$ are the decaying and damping factors, respectively. $\mathbf{h}_t^{(j)} \in \mathbb{R}^h$ is the EMA hidden state for the j-th dimension at timestep t. $\boldsymbol{\eta} \in \mathbb{R}^{d \times h}$ is the projection matrix to map the h-dimensional hidden state back to 1-dimensional output $\mathbf{y}_{t,j} \in \mathbb{R}$.

2.2 Moving Average Equipped Gated Attention

In the gated attention mechanism in MEGA, the output from EMA (1) is used to compute the shared representation (Hua et al., 2022), because it encodes contextual information through EMA. Subsequently, MEGA introduces the reset gate, the update gate , and computes the candidate activation with the update gate and the residual connection. The technical details are provided in Appendix A.

2.3 Existing Problems in MEGA

To reduce the quadratic complexity in the full attention mechanism, MEGA simply split the sequences of queries, keys and values in (14-16) into chunks of length c. The attention in (17) is individually applied to each chunk, yielding linear complexity $O(kc^2) = O(nc)$. Technically, the EMA sub-layer in MEGA helps capture local contextual information near each token, mitigating the problem of losing contextual information beyond chunk boundaries in the chunk-wise attention.

Despite the impressive successes of MEGA, it still suffers its own problems: i) the performance of MEGA with chunk-wise attention still fails behind the one with full attention, due to the limited expressiveness of the EMA sub-layer in MEGA. ii) for different tasks and/or data types, there are architectural divergences in the final MEGA architectures. For example, different normalization layers, normalization patterns (pre-norm vs. post-norm) and attention functions ($f(\cdot)$ in (17)) are applied to different data types (see Ma et al. (2023) for details). iii) There are no empirical evidences showing that MEGA is scalable for large-scale pretraining.

3 MEGALODON

To address the aforementioned problems of MEGA, in this section we describe the novel technical advancements of MEGALODON.

3.1 CEMA: Extending Multi-dimensional Damped EMA to Complex Domain

As discussed in Ma et al. (2023), the EMA component can be regarded as a simplified state space model with diagonal state matrix. Directly inspired from Gu et al. (2022b), as almost all matrices

https://doi.org/10.52202/079017-2294

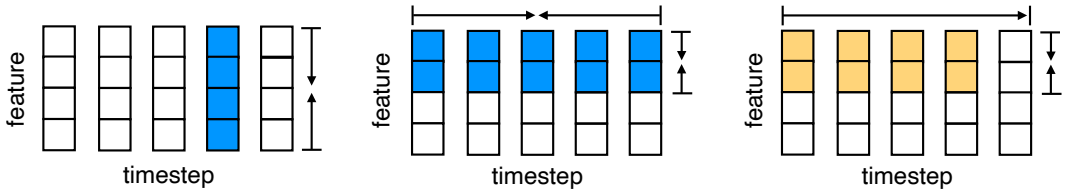


Figure 2: **Normalization methods**. The elements in blue or pink are the regions to compute means and variances. We omit the batch dimension for simplicity.

diagonalize over the complex plane, a straight-forward idea to improve EMA capability is to extend to work over the complex number system $\mathbb{C}$. We propose the *complex exponential moving average (CEMA)*, which re-writes Eq. (1):

$$\begin{aligned}\mathbf{h}_t^{(j)} &= \boldsymbol{\alpha}_j(\cos\theta_j + i\sin\theta_j) \odot \mathbf{u}_t^{(j)} + (1 - \boldsymbol{\alpha}_j \odot \boldsymbol{\delta}_j)(\cos\theta_j + i\sin\theta_j) \odot \mathbf{h}_{t-1}^{(j)} \\ \mathbf{y}_{t,j} &= \mathrm{Re}(\boldsymbol{\eta}_j^T \mathbf{h}_t^{(j)})\end{aligned} \tag{2}$$

where $\boldsymbol{\alpha}, \boldsymbol{\delta} \in \mathbb{R}^{d\times h}$ are the real number parameters same as in EMA. Different from EMA, $\boldsymbol{\eta} \in \mathbb{C}^{d\times h}$ in CEMA are complex numbers. $\theta_j \in \mathbb{R}^h,\ j \in \{1, 2, \ldots, d\}$ are the h arguments. To uniformly space the h arguments over the period 2π, we parameterize θ_j as:

$$\theta_{j,k} = \frac{2\pi k}{h}\omega_j, \quad \forall k \in \{1, 2, \ldots, h\} \tag{3}$$

where the learnable parameter $\omega \in \mathbb{R}^d$ depicts the d base angles. By decaying the absolute value of each h_t, CEMA preserves the decaying structure in kernel weights, which is a key principle to the success of convolutional models on long sequence modeling (Li et al., 2023c).

3.2 Timestep Normalization

Despite the impressive performance of Layer Normalization combined with Transformer, it is obvious that layer normalization cannot directly reduce the internal covariate shift along the spatial dimension (a.k.a timestep or sequential dimension) (Ioffe and Szegedy, 2015). Group Normalization (Wu and He, 2018) normalizes hidden representations both along the timestep dimension and a subset of the feature dimension, which has obtained improvements over Layer Normalization on a range of computer vision tasks. However, it cannot be directly applied to Transformer on auto-regressive sequence modeling, due to the leakage of future information via the mean and variance across the timestep dimension.

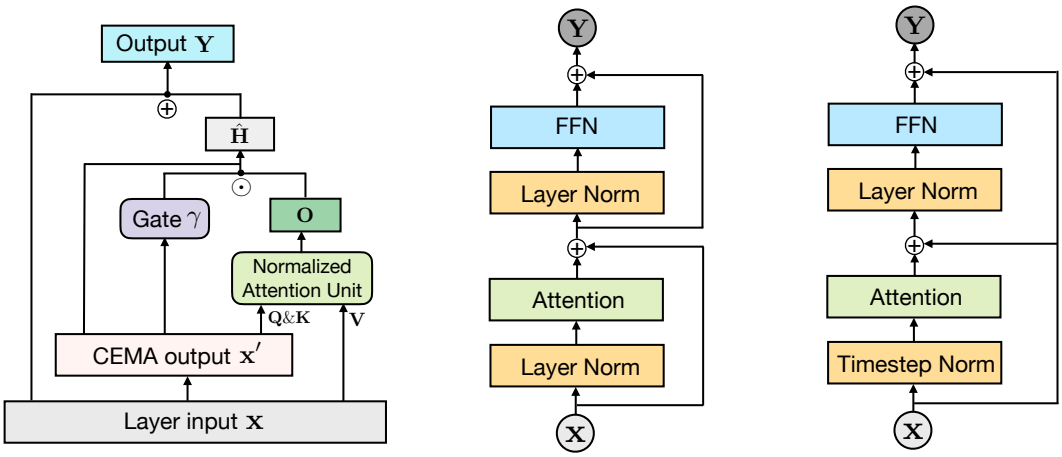
In MEGALODON, we extend Group Normalization to the auto-regressive case by computing the cumulative mean and variance. Formally, suppose an input sequence $\boldsymbol{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{n\times d}$, and k groups along the feature dimension with $d_g = d/k$ elements per group. Then, the mean and variance of the first group at timestep $t \in \{1, 2, \ldots, n\}$ are:

$$\mu_t = \frac{1}{t * d_g}\sum_{i=1}^{t}\sum_{j=1}^{d_g} x_{i,j}, \qquad \sigma_t^2 = \frac{1}{t * d_g}\sum_{i=1}^{t}\sum_{j=1}^{d_g}(x_{i,j} - \mu_t)^2 \tag{4}$$

Figure 2 illustrates Layer Normalization and Timestep Normalization. To efficiently and precisely calculate the cumulative mean and variance in each timestep, we provide hardware-friendly implementation on modern hardware (GPU) (see Appendix B.1).

3.3 Normalized Attention in MEGALODON

Previous studies have investigated the saturation and instability issues in the original scaled dot-product attention (17). A number of novel techniques have emerged to modify the scaled dot-product attention, among which normalized attention mechanisms, such as (scaled-) cosine attention (Luo et al., 2018; Liu et al., 2022) and QK-normalization (Henry et al., 2020), have stood out for the simplicity and effectiveness.



(a) Sketch of Megalodon layer. (b) Pre-Norm (c) Two-hop Residual

Figure 3: Illustration of the MEGALODON architecture. Figure (a) shows a sketch of one MEGALODON layer. Figure (b) and (c) display the configurations of pre-norm and pre-norm with two-hop residual, respectively.

Directly inspired from these normalized attention mechanisms, we propose the normalized attention mechanism specifically defined for MEGA to improve its stability. Formally,

$$\boldsymbol{X}' = \mathrm{CEMA}(\boldsymbol{X}) \qquad \in \mathbb{R}^{n\times d} \tag{5}$$

$$\boldsymbol{Z} = \boldsymbol{X}'W_z + b_z, \quad \boldsymbol{Z}' = \frac{\boldsymbol{Z}}{\|\boldsymbol{Z}\|} \qquad \in \mathbb{R}^{n\times z} \tag{6}$$

$$\boldsymbol{Q} = \boldsymbol{\kappa}_q \odot \boldsymbol{Z}' + \boldsymbol{\mu}_q \qquad \in \mathbb{R}^{n\times z} \tag{7}$$

$$\boldsymbol{K} = \boldsymbol{\kappa}_k \odot \boldsymbol{Z}' + \boldsymbol{\mu}_k \qquad \in \mathbb{R}^{n\times z} \tag{8}$$

where $\boldsymbol{Q}$ and $\boldsymbol{K}$ are computed by using the normalized shared representation $\boldsymbol{Z}'$ instead of $\boldsymbol{Z}$. Note that we remove the SiLU (Ramachandran et al., 2017) activation function ϕ_{silu} in (13), because the normalization on $\boldsymbol{Z}$ has incorporated non-linearity into $\boldsymbol{Z}'$. Then the attention operation in (17) has been changed to:

$$\boldsymbol{O} = f_{\mathrm{softmax}}\left(\boldsymbol{Q}\boldsymbol{K}^T\right)\boldsymbol{V} \qquad \in \mathbb{R}^{n\times v} \tag{9}$$

As we use learnable $\boldsymbol{\kappa}_q$, $\boldsymbol{\kappa}_k$ in (7) and (8), we can remove the scaled term $\tau(\boldsymbol{X})$. In addition, we found that with the normalized attention, the softmax function f_{softmax} obtains the best or at least comparable performance on different tasks and data modalities (see Appendix C). Hence, throughout this paper we use softmax as the default attention function.

3.4 Pre-Norm with Two-hop Residual

Normalization configurations are crucial in stably training deep architectures, and pre-normalization (Xiong et al., 2020) has become the default normalization configuration because of its better convergence properties than post-normalization in the original Transformer architecture (Vaswani et al., 2017). However, extensive studies have investigated the instability issue of pre-normalization when scaling up model size (Davis et al., 2021; Liu et al., 2022). Formally, a Transformer-based block in pre-noromalization can be formulated as (shown in Figure 3 (b)):

$$\begin{aligned}\hat{\boldsymbol{Y}} &= \mathrm{Attention}(\mathrm{Norm}(\boldsymbol{X})) + \boldsymbol{X} \\ \boldsymbol{Y} &= \mathrm{FFN}(\mathrm{Norm}(\hat{\boldsymbol{Y}})) + \hat{\boldsymbol{Y}} \\ &= \mathrm{FFN}(\mathrm{Norm}(\hat{\boldsymbol{Y}})) + \mathrm{Attention}(\mathrm{Norm}(\boldsymbol{X})) + \boldsymbol{X}\end{aligned} \tag{10}$$

where the output $\boldsymbol{Y}$ is the sum of the input $\boldsymbol{X}$ and the output of each component in one block. Hence, the range and/or variance of $\boldsymbol{Y}$ keeps increasing for deeper blocks, causing the instability issue. In

https://doi.org/10.52202/079017-2294

the original MEGA architecture, the update gate φ (19) is used for a gated residual connection (21) to mitigate this problem (Parisotto et al., 2020; Xu et al., 2020). However, the update gate φ introduces more model parameters and the instability issue still exists when scaling up model size to 7 billion.

MEGALODON introduces a new configuration named *pre-norm with two-hop residual*, which simply re-arranges the residual connections in each block (shown in Figure 3 (c):

$$\begin{aligned} \hat{\boldsymbol{Y}} &= \mathrm{Attention}(\mathrm{Norm}(\boldsymbol{X})) + \boldsymbol{X} \\ \boldsymbol{Y} &= \mathrm{FFN}(\mathrm{Norm}(\hat{\boldsymbol{Y}})) + \boldsymbol{X} \end{aligned} \tag{11}$$

where the input $\boldsymbol{X}$ is reused as the residual connection of the FFN layer. Since $\hat{\boldsymbol{Y}}$ is directly followed by a normalization layer, we remove the update gate φ and use standard residual connection. The graphical architecture of a MEGALODON sub-layer is visualized in Figure 3 (a). Note that the Timestep Normalization is only applied before the attention layer. Before the FFN layer, we still use Layer Normalization. The reasons are two-fold: i) Layer Normalization is faster than Timestep Normalization; ii) the output vector of each token from the attention layer is a mixture of vectors from contextual tokens via attention weights. Hence, normalizing the attention output along the feature dimension is similar to indirectly normalize along the timestep dimension.

3.5 4-Dimensional Parallelism in Distributed LLM Pretraining

Efficient distributed training algorithm is essential to train a large-scale language model, and several parallelization mechanisms have been introduced. The three most commonly used parallelism strategies are data, tensor (Shoeybi et al., 2019) and pipeline parallelism (Huang et al., 2019). However, the 3-dimensional parallelism is still insufficient to scale up the context length of LLMs (Li et al., 2023b; Liu et al., 2024).

Benefiting from the chunk-wise attention in MEGALODON, we can efficiently parallelize it along the new timestep/sequence dimension, which is orthogonal to all the aforementioned three parallelism dimensions. In MEGALODON, the only communications between devices in one chunk-parallel group are the last hidden state of CEMA and the cumulative mean and variance of Timestep Normalization in each block. Using asynchronous communication, we can minimize the overhead of chunk parallelization by hiding the communication costs in the computation of other components inside the same block and/or other blocks.

4 Experiments

To evaluate the scalability and efficiency of MEGALODON on long-context sequence modeling, we scale up MEGALODON to 7-billion model size and apply it to large-scale language model pretraining on 2 trillion tokens. We also conduct experiments on small/medium-scale sequence modeling benchmarks, including Long Range Arena (LRA) (Tay et al., 2021), raw speech classification on Speech Commands (Warden, 2018), image classification on ImageNet-1K (Deng et al., 2009), and language-modeling on WikiText-103 (Merity et al., 2017) and PG19 (Rae et al., 2019). [2] Empirically, MEGALODON significantly outperforms all the state-of-the-art baseline models on these tasks across various data modalities.

4.1 LLM Pretraining

Architectural Details In our MEGALODON-7B model, we adopt most of architectural hyperparameters from LLAMA2-7B to ensure fair comparison: MEGALODON-7B consists of 32 blocks, with feature dimension $d = 4096$. Following LLAMA2, we use the SwiGLU activation function (Shazeer, 2020) in the feed-forward layer, and rotary positional embedding (RoPE, Su et al. (2021)). We set the attention chunk size $c = 4096$, which is the same as the pretraining context length in LLAMA2. Benefiting from the attention gate (γ in (18)), we use a much smaller number of attention heads $h = 4$ in MEGALODON-7B, comparing to $h = 32$ in LLAMA2-7B. In addition, we apply pre-norm with two-hop residual (§3.4), using Timestep Normalization (§3.2) and Layer Normalization (Ba et al., 2016), while LLAMA2 models apply pre-normalization with RMSNorm (Zhang and Sennrich, 2019).

[2] Some results are provided in Appendix C, due to space limits.

Data and Pretraining Details We use the same mix of publicly available data from LLAMA2, ensuring that the model are trained on exactly the same 2-trillion tokens. We also use the same tokenizer as LLAMA2, whose vocabulary size is 32K.

We trained MEGALODON-7B using the AdamW optimizer (Loshchilov and Hutter, 2019), with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1e-8$. The learning rate is $3.5e-4$ and cosine learning rate schedule is applied with warmup of 2500 steps. We use a weight decay of 0.1 and gradient clipping of 1.0, and no dropout is applied during training. The context length in pretraining is 32K (4 attention chunks). The global batch size is 4M tokens, and is distributed on 256 NVIDIA A100 GPUs (16K tokens per A100). We set data parallel size to 128, chunk parallel size to 2 and tensor parallel size to 1.

Data and Computation Efficiency We evaluate the efficiency of MEGALODON w.r.t both the data and computation perspectives. For data efficiency, we display the negative log-likelihood (NLL) for MEGALODON-7B, LLAMA2-7B and LLAMA2-13B w.r.t processed tokens during training in Figure 1. MEGALODON-7B obtains significantly better (lower) NLL than LLAMA2-7B under the same amount of training tokens, demonstrating better data efficiency. Moreover, MEGALODON suffers less training spikes than the Transformer-based architecture in LLAMA2. Note that at the first 1/4 of the pretraining process ($<$ 500B tokens), the NLL of MEGALODON-7B is slightly worse than LLAMA2-7B. We found that the main reason is that we increased the base θ of RoPE from $10,000$ in LLAMA2 to $100,000$ in MEGALODON, which slows down model convergence at the beginning of the pretraining process. At the end, MEGALODON reaches a training loss of 1.70, landing mid-way between LLAMA2-7B (1.75) and LLAMA2-13B (1.67).

For computation efficiency, we conduct experiments of running LLAMA2-7B and MEGALODON-7B using the same amount of computational resources and comparing their training speed under various context lengths. Specifically, we execute each experiment to train a model with global batch size 4M tokens distributed on 256 NVIDIA A100 GPUs (16K tokens per A100) and calculate the word/token per second (WPS) to measure the training speed. Figure 4 illustrates the average WPS per device of LLAMA2-7B and MEGALODON-7B using 4K and 32K context lengths, respectively. For LLAMA2 models, we accelerate the computation of full attention with Flash-Attention V2 (Dao, 2024). Under 4K context length, MEGALODON-7B is slightly slower (about 6%) than LLAMA2-7B, due to the introduction of CEMA and Timestep Normalization. When we scale up context length to 32K, MEGALODON-7B is significantly faster (about 32%) than LLAMA2-7B, demonstrating the computation efficiency of MEGALODON for long-context pretraining. In addition, MEGALODON-7B-32K, which utilizes chunk parallelism (§3.5), achieves about 94% utilization of MEGALODON-7B-4K.

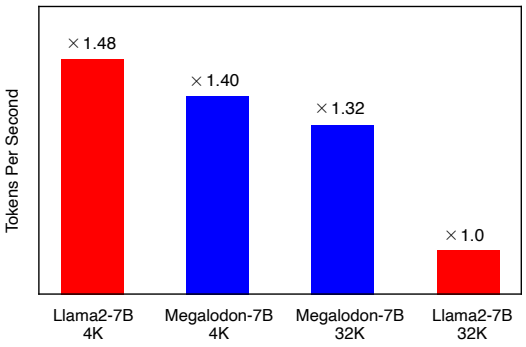


Figure 4: Average WPS per device.

4.2 Short-Context Evaluation on Academic Benchmarks

We compare MEGALODON-7B to LLAMA2 models on standard academic benchmarks with short contexts ($<$ 4K tokens), closely following the settings in LLAMA2 (Touvron et al., 2023). The benchmarks are grouped into the categories listed below:

- **Commonsense Reasoning** (0-shot): HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-e and -c (Clark et al., 2018).
- **World Knowledge** (5-shot): NaturalQuestions (NQ, Kwiatkowski et al. (2019)) and TriviaQA (TQA, Joshi et al. (2017)).
- **Reading Comprehension** (0-shot): BoolQ (Clark et al., 2019).
- **Popular aggregated results** (5-shot): MMLU (Hendrycks et al., 2020).

Table 1 summarizes the results of MEGALODON and LLAMA2 on these academic benchmarks, together with other open-source base models, including MPT (MosaicML, 2023), RWKV (Peng et al., 2023), Mamba (Gu and Dao, 2023), Mistral (Jiang et al., 2023) and Gemma (Mesnard et al., 2024). Pretrained on the same 2T tokens, MEGALODON-7B surpasses LLAMA2-7B across all the

https://doi.org/10.52202/079017-2294

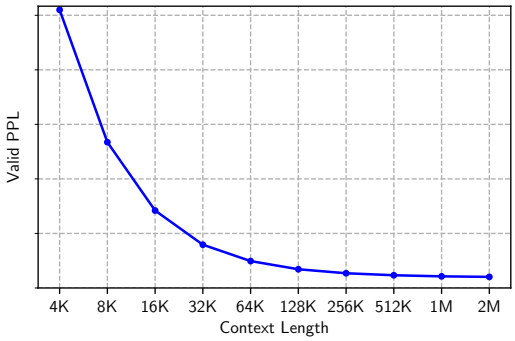


Figure 5: PPL in various context lengths.

Model	NaQA	Qasper	QMSum
Xgen	17.4	20.5	6.8
MPT	18.8	24.7	8.8
Yarn	20.9	26.2	11.4
LLAMA2	18.8	19.8	10.1
LLAMA2-L*	23.5	**28.3**	**14.5**
MEGALODON	**23.9**	28.0	13.1

Table 2: **Results on Scrolls**. * LLAMA2-L (Xiong et al., 2023) continually trains LLAMA2 on 500B tokens for length extension.

benchmarks. On some tasks, MEGALODON-7B achieves comparable or even better performance with LLAMA2-13B. Note that Mistral-7B and Gemma-8B were pretrained on much larger datasets than MEGALODON-7B, hence the results are not directly comparable.

4.3 Long-Context Evaluation

Perplexity over Long Sequences To demonstrate the capability of MEGALODON to make use of very long contexts to improve next-token prediction, we start by conducting the evaluation of valid perplexity on different context lengths. Concretely, we construct a validation dataset which consists of 1,920 selected books. Each of these books contains sequences with at least 2M tokens. The validation dataset is constructed by first randomly shuffling all the files and then concatenating them. Figure 5 shows the perplexity (PPL) of the validation dataset in various context lengths ranging from 4K to 2M. We observe that the PPL decreases monotonically with context length, validating the effectivenss and robustness of MEGALODON on modeling extremely long sequences.

Long-Context QA tasks in Scrolls Next, we evaluate MEGALODON on long-context open-book question answering (QA) tasks in the Scrolls dataset (Shaham et al., 2022), including NarrativeQA (Kočiský et al., 2018), Qasper (Dasigi et al., 2021) and QMSum (Zhong et al., 2021). Following Xiong et al. (2023), we use a simple prompt {CONTEXT} Q: {QUESTION} A: for all the tasks, and evaluate 0-shot F1-score on NarrativeQA, 2-shot F1-score on Qasper and 1-shot geometric-ROUGE[3] on QMSum. Table 2 lists the results of MEGALODON-7B, together with other open-source long-context models in the scale of 7B, namely Xgen-7B-8K (Nijkamp et al., 2023), MPT-7B-8K (MosaicML, 2023), YaRN-7B-128k (Peng et al., 2024), LLAMA2-7B-4K (Touvron et al., 2023) and LLAMA2-7B-32K (LLAMA2-L, Xiong et al. (2023)). MEGALODON-7B obtains the best F1 on NarrativeQA, and competitive results with LLAMA2-7B Long. It should be noticed that LLAMA2-7B Long extends the context length of LLAMA2-7B from 4K to 32K by continually pretraining it on additional 500B tokens from long-context data.

4.4 Instruction Finetuning

To evaluation the generalization capability of MEGALODON on instruction following and alignment, We finetune the base model of MEGALODON-7B on a proprietary instruction-alignment data under a controlled setting. We did not apply any RLHF techniques to further finetune it. Table 3 summarizes the performance of chat models in 7B scale on MT-Bench[4]. MEGALODON exhibits superior performance on MT-Bench compared to Vicuna (Chiang et al., 2023), and comparable performance to LLAMA2-Chat, which utilizes RLHF for further alignment finetuning. We present some outputs from instruction finetuned MEGALODON in Appendix D.

Table 3: **MT Bench**. Comparison of Chat models. * LLAMA2-Chat utilizes RLHF.

Model	Size	MT-Bench
Vicuna	7B	6.17
LLAMA2-Chat*	7B	6.27
Mistral-Instruct	7B	6.84
MEGALODON	7B	6.27

[3]Geometric mean of ROUGE-1, 2 and L.

[4]https://klu.ai/glossary/mt-bench-eval

Table 4: (**ImageNet-1K**) Top-1 accuracy.

Model	#Param.	Acc.
ResNet-152	60M	78.3
ViT-B	86M	77.9
DeiT-B	86M	81.8
MEGA	90M	82.3
MEGALODON	90M	**83.1**

Table 5: (**PG-19**) Word-level perplexity.

Model	#Param.	Val	Test
Compressive Trans.	–	43.4	33.6
Perceiver AR	975M	45.9	28.9
Block-Recurrent Trans.	1.3B	–	26.5
MEGABYTE	1.3B	42.8	36.4
MEGALODON	1.3B	**29.5**	**25.4**

4.5 Evaluation on Medium-Scale Benchmarks

ImageNet Classification To evaluate MEGALODON on image classification task, we conduct experiments on the Imagenet-1K (Deng et al., 2009) dataset, which consists of 1.28M training images and 50K validation images from 1000 classes. We mostly follow DeiT's approach of applying several data augmentation and regularization methods that facilitate the training process, and adopt most the hyperparameters from Ma et al. (2023). For classification task, we replace the timestep normalization with the standard group normalization method. Top-1 accuracy on the validation set is reported in Table 4 to assess various models. MEGALODON obtains about 1.3% accuracy improvement over DeiT-B (Touvron et al., 2021), and 0.8%. improvement over MEGA (Ma et al., 2023).

Auto-regressive Language Modeling on PG-19 We also evaluate MEGALODON on auto-regressive language modeling on the medium-scale PG19 (Rae et al., 2019) datasets. We use the same vocabulary from Block-Recurrent Transformer (Hutchins et al., 2022) and adopt most of its hyper-parameters to train a MEGALODON model with 1.3B parameters. Table 5 illustrate the word-level perplexity (PPL) of MEGALODON on PG-19, together with previous state-of-the-art models, including Compressive Transformer (Rae et al., 2020), Perceiver AR (Hawthorne et al., 2022), Block-Recurrent Transformer (Hutchins et al., 2022) and MEGABYTE (Yu et al., 2024). MEGALODON significantly outperforms all the baselines.

5 Conclusion

We have introduced MEGALODON, an improved MEGA architecture with multiple novel technical components, including complex exponential moving average (CEMA), the timestep normalization layer, normalized attention and pre-norm with two-hop residual configuration, to improve its capability, efficiency and scalability. Through a direct comparison with LLAMA2, MEGALODON achieves impressive improvements on both training perplexity and across downstream benchmarks. Importantly, experimental results on long-context modeling demonstrate MEGALODON's ability to model sequences of unlimited length. Additional experiments on small/medium-scale benchmarks across different data modalities illustrate the robust improvements of MEGALODON, which lead to a potential direction of future work to apply MEGALODON for large-scale multi-modality pretraining.

Acknowledgments

We thank Sadhika Malladi, Zihao Ye, Dacheng Li and Rulin Shao for their helpful feedback and discussion during this work.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*, 2018.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7432–7439, 2020.

https://doi.org/10.52202/079017-2294

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR-2024)*, 2024.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-2021)*, pages 4599–4610, Online, June 2021. Association for Computational Linguistics.

Jared Q Davis, Albert Gu, Krzysztof Choromanski, Tri Dao, Christopher Re, Chelsea Finn, and Percy Liang. Catformer: Designing stable transformers via sensitivity analysis. In *International Conference on Machine Learning*, pages 2489–2499. PMLR, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Daniel Y Fu, Tri Dao, Khaled Kamal Saab, Armin W Thomas, Atri Rudra, and Christopher Re. Hungry hungry hippos: Towards language modeling with state space models. In *The Eleventh International Conference on Learning Representations (ICLR-2023)*, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations (ICLR-2022)*, 2022a.

Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *arXiv preprint arXiv:2206.11893*, 2022b.

Stephen Hanson and Lorien Pratt. Comparing biases for minimal network construction with back-propagation. *Advances in neural information processing systems*, 1, 1988.

Curtis Hawthorne, Andrew Jaegle, Cătălina Cangea, Sebastian Borgeaud, Charlie Nash, Mateusz Malinowski, Sander Dieleman, Oriol Vinyals, Matthew Botvinick, Ian Simon, et al. General-purpose, long-context autoregressive modeling with perceiver ar. In *International Conference on Machine Learning*, pages 8535–8558. PMLR, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4246–4253, 2020.

Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In *International Conference on Machine Learning (ICML-2022)*, pages 9099–9117. PMLR, 2022.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, and zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

J Stuart Hunter. The exponentially weighted moving average. *Journal of quality technology*, 18(4): 203–210, 1986.

DeLesley Hutchins, Imanol Schlag, Yuhuai Wu, Ethan Dyer, and Behnam Neyshabur. Block-recurrent transformers. *Advances in neural information processing systems*, 35:33248–33261, 2022.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning (ICML-2015)*, pages 448–456. pmlr, 2015.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017.

William Kahan. Pracniques: further remarks on reducing truncation errors. *Communications of the ACM*, 8(1):40, 1965.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *Technical Report. University of Toronto*, 2009.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Bonan Li, Yinhan Hu, Xuecheng Nie, Congying Han, Xiangjian Jiang, Tiande Guo, and Luoqi Liu. Dropkey for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22700–22709, June 2023a.

Dacheng Li, Rulin Shao, Anze Xie, Eric P Xing, Joseph E Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. Lightseq:: Sequence level parallelism for distributed training of long context transformers. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ NeurIPS 2023)*, 2023b.

Yuhong Li, Tianle Cai, Yi Zhang, Deming Chen, and Debadeepta Dey. What makes convolutional models great on long sequence modeling? In *International Conference on Learning Representations (ICLR-2023)*, 2023c.

Drew Linsley, Junkyung Kim, Vijay Veerabadran, Charles Windolf, and Thomas Serre. Learning long-range spatial dependencies with horizontal gated recurrent units. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. In *International Conference on Learning Representations (ICLR-2024)*, 2024.

 https://doi.org/10.52202/079017-2294

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Chunjie Luo, Jianfeng Zhan, Xiaohe Xue, Lei Wang, Rui Ren, and Qiang Yang. Cosine normalization: Using cosine similarity instead of dot product in neural networks. In *27th International Conference on Artificial Neural Networks (ICANN-2018)*, pages 382–391. Springer, 2018.

Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. Luna: Linear unified nested attention. *Advances in Neural Information Processing Systems*, 34:2441–2453, 2021.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. Mega: Moving average equipped gated attention. In *The Eleventh International Conference on Learning Representations*, 2023.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR-2017)*, 2017.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Pier Giuseppe Sessa, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024.

MosaicML. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023.

Nikita Nangia and Samuel Bowman. Listops: A diagnostic dataset for latent tree learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 92–99, 2018.

Erik Nijkamp, Tian Xie, Hiroaki Hayashi, Bo Pang, Congying Xia, Chen Xing, Jesse Vig, Semih Yavuz, Philippe Laban, Ben Krause, Senthil Purushwalkam, Tong Niu, Wojciech Kryściński, Lidiya Murakhovs'ka, Prafulla Kumar Choubey, Alex Fabbri, Ye Liu, Rui Meng, Lifu Tu, Meghana Bhat, Chien-Sheng Wu, Silvio Savarese, Yingbo Zhou, Shafiq Joty, and Caiming Xiong. Xgen-7b technical report, 2023.

Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pages 7487–7498. PMLR, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *International Conference on Learning Representations (ICLR-2024)*, 2024.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International conference on machine learning (ICML-2023)*. PMLR, 2023.

Dragomir R Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. The acl anthology network corpus. *Language Resources and Evaluation*, 47(4):919–944, 2013.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint*, 2019. URL https://arxiv.org/abs/1911.05507.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. Compressive transformers for long-range sequence modeling. In *International Conference on Learning Representations (ICLR-2020)*, 2020.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 7(1):5, 2017.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. SCROLLS: Standardized CompaRison over long language sequences. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP-2022)*, pages 12007–12021, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qVyeW-grC2k.

Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Won Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Q. Tran, Dani Yogatama, and Donald Metzler. Scaling laws vs model architectures: How does inductive bias influence scaling?, 2022.

 https://doi.org/10.52202/079017-2294

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. Beyond the limits: A survey of techniques to extend the context length in large language models, 2024.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV-2018)*, pages 3–19, 2018.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling of foundation models. *arXiv preprint arXiv:2309.16039*, 2023.

Hongfei Xu, Qiuhui Liu, Deyi Xiong, and Josef van Genabith. Transformer with depth-wise lstm. *arXiv preprint arXiv:2007.06257*, 2020.

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. Megabyte: Predicting million-byte sequences with multiscale transformers. *Advances in Neural Information Processing Systems*, 36, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL-2019)*. Association for Computational Linguistics, 2019.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-2021)*, pages 5905–5921, Online, June 2021. Association for Computational Linguistics.

Yongchao Zhou, Uri Alon, Xinyun Chen, Xuezhi Wang, Rishabh Agarwal, and Denny Zhou. Transformers can achieve length generalization but not robustly, 2024.

Appendix: MEGALODON: Efficient Long-Context LLM Pretraining and Inference with Unlimited Context Length

A Background: Moving Average Equipped Gated Attention

In the gated attention mechanism in MEGA, the output from EMA (1) is used to compute the shared representation (Hua et al., 2022) $\boldsymbol{Z}$:

$$\boldsymbol{X}' = \text{EMA}(\boldsymbol{X}) \qquad \in \mathbb{R}^{n \times d} \tag{12}$$

$$\boldsymbol{Z} = \phi_{\text{silu}}(\boldsymbol{X}' W_z + b_z) \qquad \in \mathbb{R}^{n \times z} \tag{13}$$

where $\boldsymbol{X}'$ can be regarded as the updated or contextual input, because it encodes contextual information through EMA. Then, the query and key sequences are computed by applying per-dimension scalars and offsets to $\boldsymbol{Z}$, and the value sequence is from the original $\boldsymbol{X}$:

$$\boldsymbol{Q} = \boldsymbol{\kappa}_q \odot \boldsymbol{Z} + \boldsymbol{\mu}_q \qquad \in \mathbb{R}^{n \times z} \tag{14}$$

$$\boldsymbol{K} = \boldsymbol{\kappa}_k \odot \boldsymbol{Z} + \boldsymbol{\mu}_k \qquad \in \mathbb{R}^{n \times z} \tag{15}$$

$$\boldsymbol{V} = \phi_{\text{silu}}(\boldsymbol{X} W_v + b_v) \qquad \in \mathbb{R}^{n \times v} \tag{16}$$

where $\boldsymbol{\kappa}_q, \boldsymbol{\mu}_q, \boldsymbol{\kappa}_k, \boldsymbol{\mu}_k \in \mathbb{R}^z$ are the learnable scalars and offsets of queries and keys, respectively. v is the expanded intermediate dimension for the value sequence. The output of attention is computed as follows:

$$\boldsymbol{O} = f\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\tau(\boldsymbol{X})}\right)\boldsymbol{V} \qquad \in \mathbb{R}^{n \times v} \tag{17}$$

Subsequently, MEGA introduces the reset gate $\boldsymbol{\gamma}$, the update gate $\boldsymbol{\varphi}$, and computes the candidate activation $\hat{\boldsymbol{H}}$ and final output $\boldsymbol{Y}$:

$$\boldsymbol{\gamma} = \phi_{\text{silu}}(\boldsymbol{X}' W_\gamma + b_\gamma) \qquad \in \mathbb{R}^{n \times v} \tag{18}$$

$$\boldsymbol{\varphi} = \phi_{\text{sigmoid}}(\boldsymbol{X}' W_\varphi + b_\varphi) \qquad \in \mathbb{R}^{n \times d} \tag{19}$$

$$\hat{\boldsymbol{H}} = \phi_{\text{silu}}(\boldsymbol{X}' W_h + (\boldsymbol{\gamma} \odot \boldsymbol{O}) U_h + b_h) \qquad \in \mathbb{R}^{n \times d} \tag{20}$$

$$\boldsymbol{Y} = \boldsymbol{\varphi} \odot \hat{\boldsymbol{H}} + (1 - \boldsymbol{\varphi}) \odot \boldsymbol{X} \qquad \in \mathbb{R}^{n \times d} \tag{21}$$

with the update gate $\boldsymbol{\varphi}$ and the residual connection $\boldsymbol{X}$.

B Implementation Details

B.1 Efficient Fused CUDA Operators Implementation

Fused Attention We implemented a fused attention operator to improve the efficiency, mainly by fusing the causal mask, softmax function and dropout operation (if necessary). The fused implementation reduces the IO costs from global memory for the attention weight. For attention dropout, we adopt the dropout-before-softmax scheme in DropKey (Li et al., 2023a), which applies the dropout mask on the input attention matrix of the softmax function. Concretely, we fill the values of the attention matrix at dropout mask positions to $-\infty$ before feeding it into the softmax function. One important advantage of this dropout-before-softmax scheme comparing to the standard attention dropout is that the computation of the gradients in back-propagation is independent with the applied dropout mask.

Efficient FFTConv We also provide an efficient fused implementation of the FFTConv operator. Similar with the FlashConv in H3 (Fu et al., 2023), we fused the real number FFT (RFFT), its inverse (IRFFT) and implemented the Cooley-Tukey FFT algorithm (Cooley and Tukey, 1965) in the CUDA shared memory. Similar with the FlashConv in H3 (Fu et al., 2023), we fused the real number FFT (RFFT), its inverse (IRFFT) and the element-wise multiplication, and implemented the Cooley-Tukey FFT algorithm (Cooley and Tukey, 1965) in CUDA's shared memory. Our implementation is able to accommodate up to 16K tokens in the limited shared memory of an A100 GPU.

https://doi.org/10.52202/079017-2294

Table 6: (**Long Range Arena**) Accuracy on the full suite of long range arena (LRA) tasks. Results of previous models are reported in Ma et al. (2023).

Models	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg.
Transformer	37.11	65.21	79.14	42.94	71.83	✗	59.24
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.67
Linformer	35.70	53.94	52.27	38.56	76.34	✗	51.36
BigBird	36.05	64.02	59.29	40.83	74.87	✗	55.01
Luna-256	37.98	65.78	79.56	47.86	78.55	✗	61.95
S4	59.10	86.53	90.94	88.48	94.01	96.07	85.86
MEGA-chunk	58.76	90.19	90.97	85.80	94.41	93.81	85.66
MEGA	63.14	90.43	91.25	**90.44**	96.01	97.98	88.21
MEGALODON-chunk	62.23	**90.53**	91.74	87.11	96.89	97.21	87.62
MEGALODON	**63.79**	90.48	**91.76**	89.42	**98.13**	**98.17**	**88.63**

Timestep Normalization For the TimestepNorm operator, we have an efficient implementation to improve both its speed and numerical stability. To compute the cumulative mean and variance for each of the timesteps, our implementation distributed the threads in each CUDA block in both the timestep/sequence dimension and the feature dimension to balance the parallelism of the algorithm and the performance of the global memory access. To improve numerical stability, we used the Welford algorithm (Welford, 1962) to compute the cumulative mean and variance and the Kahan Summation (Kahan, 1965) to reduce the numerical error from summation.

B.2 Plus 1 Reparameterization in Normalization Layers

In the normalization methods, two learnable parameters γ and β are introduced to scale and shift the normalized value:

$$y = \gamma \frac{x - \mu}{\sigma} + \beta \tag{22}$$

where μ and σ^2 are the mean and variance of the input x across the pre-defined dimensions. Initialization of γ and β is crucial for model performance and stability. The standard implementation of normalization layers, such as PyTorch (Paszke et al., 2019), initializes γ and β to vectors of ones and zeros, respectively, to preserve the mean and variance of the normalized inputs at the beginning of training.

This standard implementation, however, suffers a problem when weight decay regularization is applied to prevent overfitting (Hanson and Pratt, 1988). Technically, the weight decay regularization pushes the values of model parameters towards smaller magnitudes. In the context of normalization methods, weight decay pushes the values in γ towards zero, which diverges from its initialization of one. This may prevent the model from learning the true scale of the data distribution, and may cause numerical stability issues as well.

To address this problem, we used the *plus 1 reparameterization*[5] of the scale parameter γ:

$$y = (\gamma + 1) \frac{x - \mu}{\sigma} + \beta \tag{23}$$

where γ is initialized to zero. Under weight decay, γ remains centered around zero, resulting in a desirable scale of $\gamma + 1$ around one.

C Experiments on Small-Scale Benchmarks

We conducted small-scale experiments on five benchmarks across various data modalities, including text, audio and image. To demonstrate the robustness of the MEGALODON architecture on different tasks and data types, we used a single unified architecture with minimal architectural divergence in

[5]Similar idea in the blog: https://medium.com/@ohadrubin/exploring-weight-decay-in-layer-normalization-challenges-and-a-reparameterization-solution-ad4d12c24950

Table 7: (**SC-Raw**) Accuracy.

Model	#Param.	Acc.
Transformer	786K	✗
S4	300K	97.50
MEGA	300K	96.92
MEGA (big)	476K	97.30
MEGALODON	300K	**98.14**

Table 8: (**WikiText-103**) Word-level PPL.

Model	#Param.	PPL
Transformer	247M	18.66
Transformer-XL	257M	18.30
S4	249M	20.95
MEGA	252M	18.07
MEGALODON	252M	**17.23**

all the experiments: softmax attention function, rotary postional embedding, pre-norm with two-hop residual, and timestep Normalization (Group Normalization for classification). We adopt (almost) all the architectural and training hyperparameters from the corresponding experiments of the original MEGA (Ma et al., 2023).

C.1 Long Range Arena (LRA)

Long Range Arena (LRA) benchmark (Tay et al., 2021) is designed for evaluating sequence models under the long-context scenario. They collect six tasks in this benchmark which are ListOps (Nangia and Bowman, 2018), byte-level text classification (Text; Maas et al. (2011)), byte-level document retrieval (Retrieval; Radev et al. (2013)), image classification on sequences of pixels (Image; Krizhevsky et al. (2009)), Pathfinder (Linsley et al., 2018) and its extreme long version (Path-X; Tay et al. (2021)). These tasks consist of input sequences ranging from 1K to 16K tokens and span across a variety of data types and modalities.

Table 6 compares MEGALODON against several baselines, including Transformer and its efficient variants, the state space model S4 (Gu et al., 2022a), and the original MEGA model. Following Ma et al. (2023), we also evaluate MEGALODON-chunk on each task, by setting the chunk size $c = 128$ for all the tasks, except Path-X where $c = 4096$. With chunk-wise attention, MEGALODON-chunk substantially outperforms MEGA-chunk on all the six tasks. In addition, MEGALODON significantly narrows the gap between chunk-wise attention and full attention.

C.2 Raw Speech Classification

To evaluate the capability of MEGALODON on the long-range modeling of speech signals, we apply MEGALODON to classify raw speech (with length 16000), rather than using traditional preprocessing (e.g. convert to MFCC features). Following Ma et al. (2023), we perform speech classification on the SC10 subset of the Speech Commands dataset (Warden, 2018), and set attention chunk size $c = 1000$. As shown in Table 7, our MEGALODON model with 300K parameters achieves an accuracy of 98.14, which is significantly better than 97.50 from S4 and 96.92 from MEGA with similar model size.

C.3 Auto-regressive Language Modeling

We also evaluate MEGALODON on auto-regressive language modeling on the small-scale WikiText-103 (Merity et al., 2017) datasets. Following Baevski and Auli (2018), we adopt adaptive softmax and input embeddings and use a vocabulary of 260K tokens for WikiText-103 and train a MEGALODON model with about 250M parameters with attention chunk size $c = 2048$. Table 8 illustrate the word-level perplexity (PPL) of MEGALODON on WikiText-103, together with previous state-of-the-art models, including Transformer (Baevski and Auli, 2018), Transformer-XL (Dai et al., 2019), S4 (Gu et al., 2022a) and MEGA (Ma et al., 2023). MEGALODON significantly outperforms all the baselines on both the two datasets.

D Model Outputs from Instruction-finetuned MEGALODON

https://doi.org/10.52202/079017-2294

What is the central dogma of molecular biology? What processes are involved? Who named this?

The central dogma of molecular biology describes the flow of genetic information from DNA to RNA to proteins. The three processes involved in the central dogma are transcription, translation, and translocation.
The central dogma was first proposed by Francis Crick, James Watson, and Maurice Wilkins in the 1950s, and was later formalized by Crick in a 1958 paper titled "On the Central Dogma of Molecular Biology". The term "central dogma" was coined by Crick to describe the flow of genetic information from DNA to RNA to proteins.

Identify and fix one incorrect fact in your previous response.

The correct name of the person who proposed the central dogma of molecular biology is James Watson, not Maurice Wilkins. James Watson, along with Francis Crick, first proposed the central dogma in the 1950s, and Maurice Wilkins was a co-recipient of the Nobel Prize in Physiology or Medicine in 1962 for his contributions to the discovery of the double helix structure of DNA.

Act as a math teacher. I will provide some mathematical equations or concepts, and it will be your job to explain them in easy-to-understand terms. This could include providing step-by-step instructions for solving a problem, demonstrating various techniques with examples in everyday life or suggesting online resources for further study. My first request is "I need help understanding how probability works."

Probability is a branch of mathematics that deals with the likelihood of an event occurring. It is often expressed as a number between 0 and 1, where 0 represents an impossible event and 1 represents a certain event.
For example, if you were to roll a fair six-sided die, the probability of rolling a 1 would be 1/6, or about 17%. This is because there is only one way for the die to land on a 1 out of the six possible outcomes.
To calculate probability, you can use the formula:
Probability = Number of ways the event can occur / Total number of possible outcomes
For example, if you were to ask "What is the probability of flipping heads on a fair coin toss?", the answer would be 1/2, or 50%. This is because there is only one way for the coin to land heads-up out of the two possible outcomes (heads or tails).
There are many different ways to interpret probability, and it can be used in a variety of real-world applications such as gambling, insurance, and statistics.
I hope this helps to give you a basic understanding of probability. Let me know if you have any questions or need further clarification.

What are the differences between Riemannian geometry and euclidean geometry?

Riemannian geometry and Euclidean geometry are both branches of mathematics that study the properties of space, but they differ in their assumptions and methods.
Euclidean geometry is based on the assumption that the space is flat and flat Euclidean space is the simplest non-empty connected flat metric space. It studies the properties of space using concepts such as distance, angle, and parallelism. It is named after the Greek mathematician Euclid, who developed the foundations of the field in his book "Elements".
Riemannian geometry, on the other hand, is based on the assumption that the space is curved. It studies the properties of space using concepts such as distance, angle, and parallelism, but in a more general setting, and it is named after the German mathematician Bernhard Riemann.
In Riemannian geometry, the concept of a metric is used to measure the distance between two points in a curved space. A metric is a mathematical function that assigns a distance to each segment in a space. The most common metric in Riemannian geometry is the distance function induced by a Riemannian metric, which is a mathematical function that assigns a distance to each segment in a space.
In summary, Euclidean geometry is a special case of Riemannian geometry, where the metric is the flat Euclidean metric. Riemannian geometry is a more general field that studies the properties of curved spaces, and it is used in many areas of mathematics, physics and other sciences.
I hope this helps to give you a basic understanding of the differences between Riemannian geometry and Euclidean geometry. Let me know if you have any questions or need further clarification.

Evaluate the following movie reviews on a scale of 1 to 5, with 1 being very negative, 3 being neutral, and 5 being very positive: 1. This movie released on Nov. 18, 2019, was phenomenal. The cinematography, the acting, the plot - everything was top-notch. 2. Never before have I been so disappointed with a movie. The plot was predictable and the characters were one-dimensional. In my opinion, this movie is the worst one to have been released in 2022. 3. The movie was okay. There were some parts I enjoyed, but there were also parts that felt lackluster. This is a movie that was released in Feb 2018 and seems to be quite ordinary. Return the answer as a JSON array of integers.

[5, 1, 3]

Update your previous reply by including the release date as part of the JSON content.

```
[
    {"release_date": "Nov 18, 2019",    "rating": 5},
    {"release_date": "2022", "rating": 1},
    {"release_date": "Feb 2018", "rating": 3}
]
```

Figure 6: Model outputs from instruction-finetuned MEGALODON.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: see the abstract

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: No theoretical results involved.

https://doi.org/10.52202/079017-2294

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: see appendix

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: see appendix

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: see appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).

https://doi.org/10.52202/079017-2294

- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: see experiments and appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

https://doi.org/10.52202/079017-2294

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.